ALBERT: A Lite BERT for Language Understanding

张老师

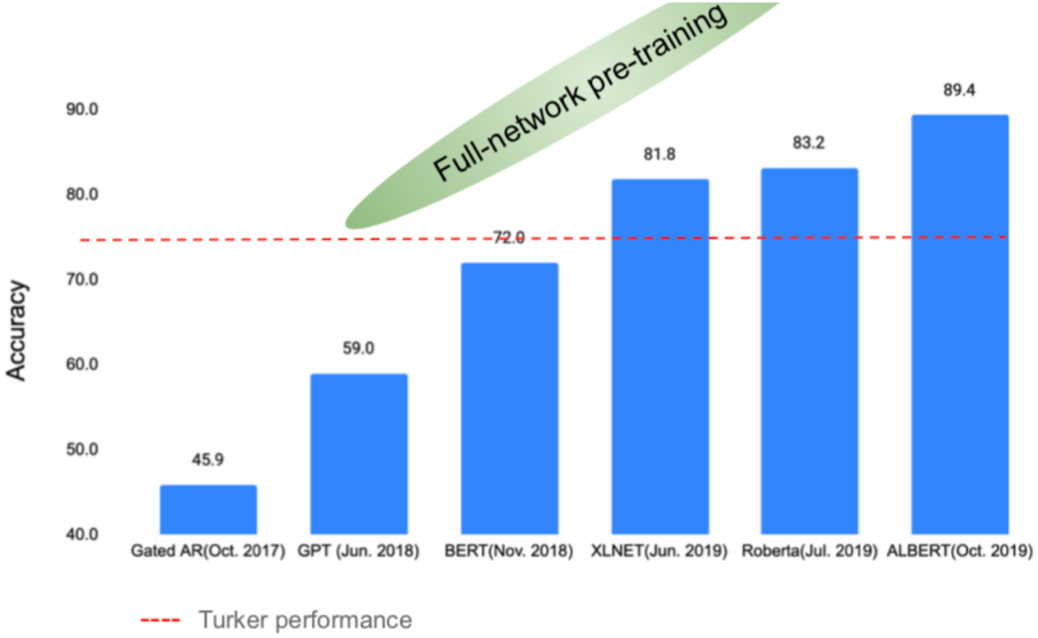
2020|09|20

Slide credit: Kaiming He, Zhenzhong Lan

GREEDY TECHNOLOGY 贪心科技
GREEDY TECHNOLOGY 贪心科技

- We are witnessing a big shift in the approach in natural language understanding in the last two years.
- Full-network pre-training share most of the parameters between pre-training and fine-tuning: GPT, Bert……
- Full-network pre-training is the AlexNet in NLU

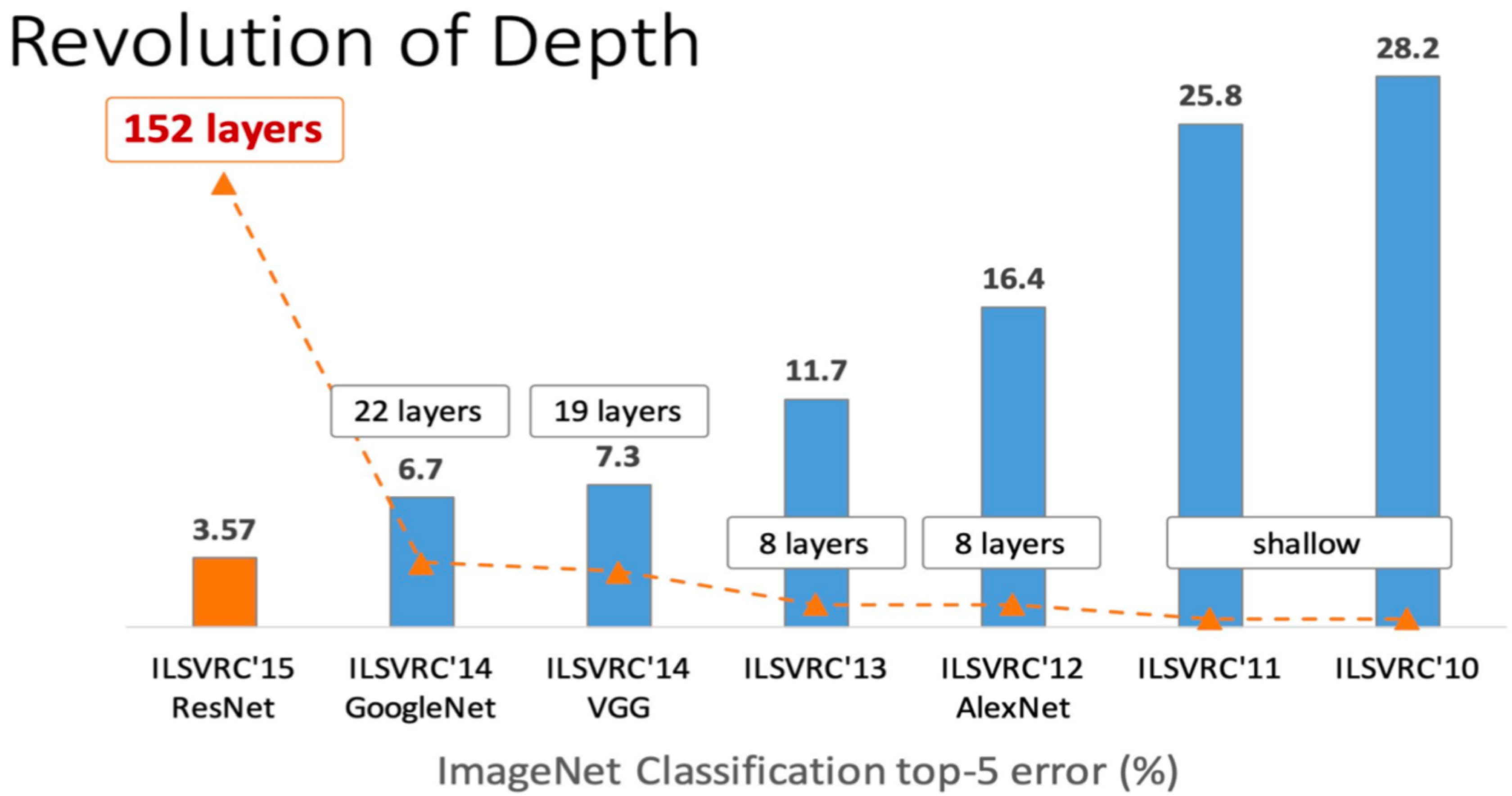
Improvements on ImageNet over the years



Improvements on RACE in the last two years

RACE: English Reading Comprehension Exam for Chinese Middle/High School

- Can we improve Full-network pre-training models, similar to what computer vision community did for AlexNet?



For BERT, increasing the depth (L) and width (H) of the network leads to better performance

Hyperparams			Dev Set Accuracy			
#L	#H	#A	LM (ppl)	MNLI-m	MRPC	SST-2
3	768	12	5.84	77.9	79.8	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	768	12	3.99	84.4	86.7	92.9
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7

Larger network

Better performance

Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." 2018

- Is having better NLU model as easy as increasing the model size?
- Not quite！

Increasing model size leads to **parameter explosion** and become difficult to train on TPUs

Model	Hidden Size	Parameters	RACE (Accuracy)
BERT-large (Devlin et al., 2019)	1024	334M	72.0%
BERT-large (ours)	1024	334M	73.9%
BERT-xlarge (ours)	2048	1270M	

Hit the memory limitation of GPUs/TPUs

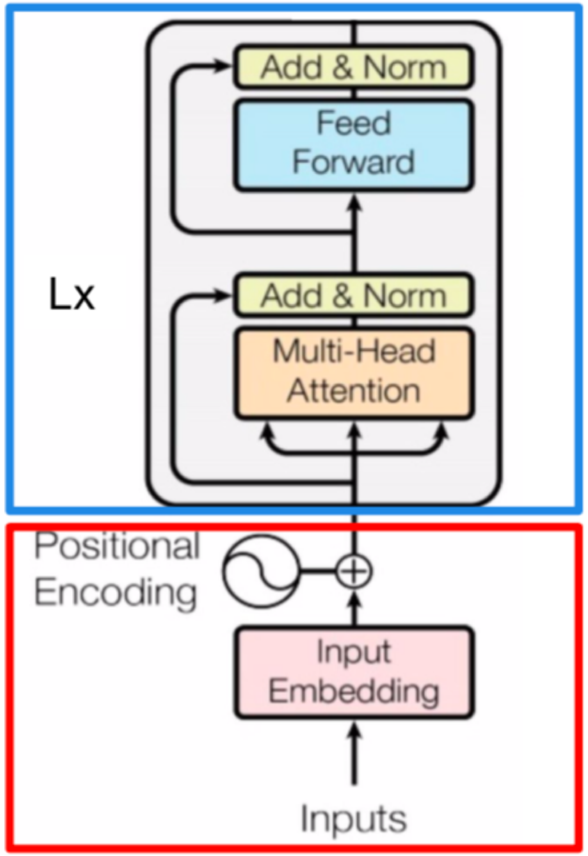
- Can we have more efficient ways of using parameters?
- Where we have the most parameters from?

Attention feed-forward block

- Parameter size O (12 x L x H x H)
- L: layer size
- H: hidden size -- needs to be large, e.g., ≥ 768

Token embedding projection block

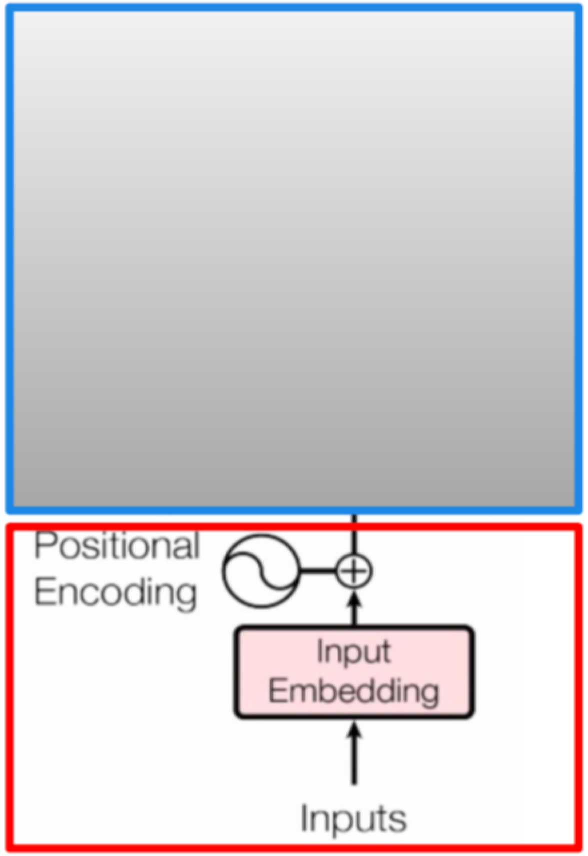
- Parameter size O (V x E)
- V: vocab size -- needs to be large, e.g., 30K
- E: embedding size -- often the same size as H.



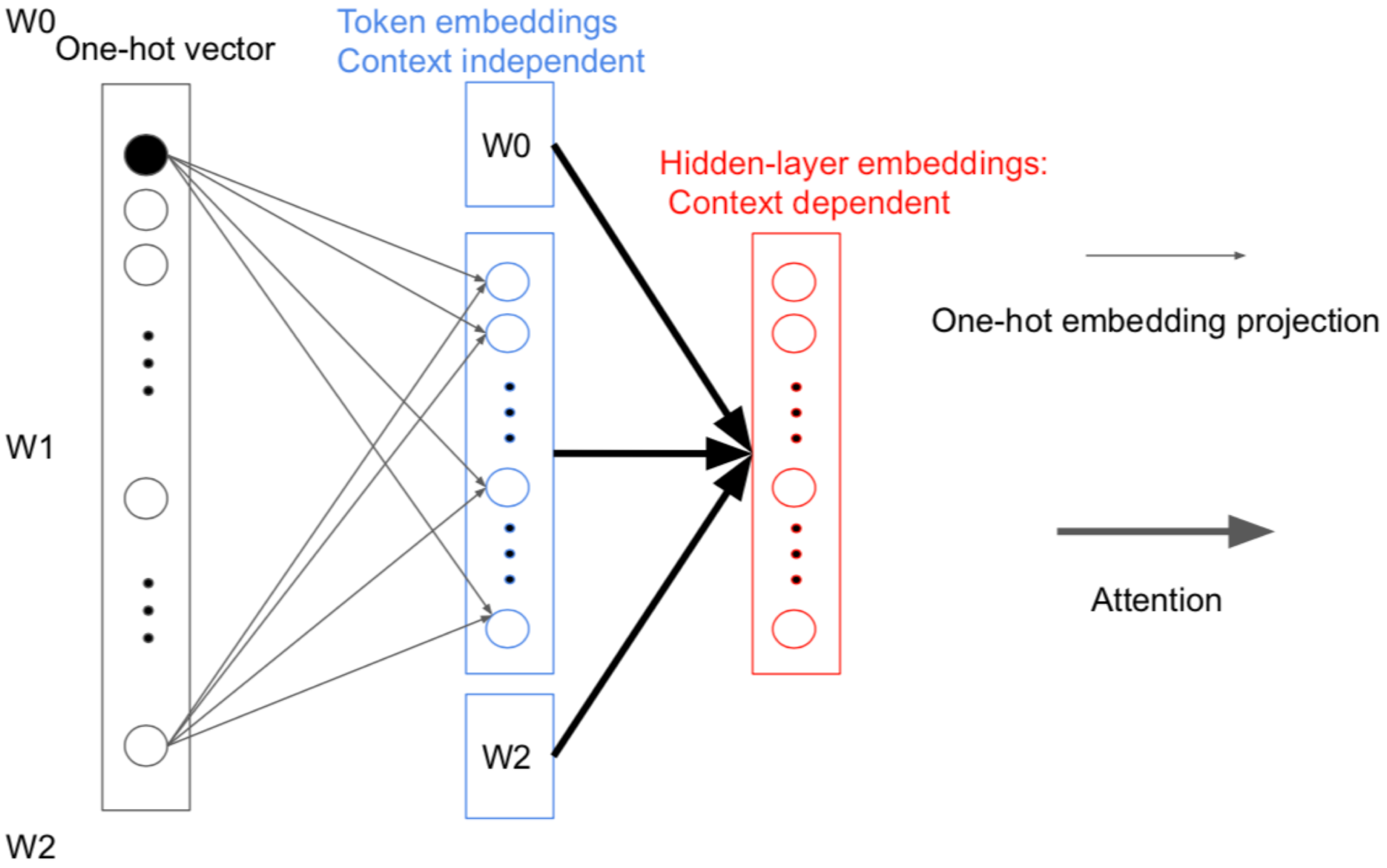
Method 1: factorized embedding parametrization

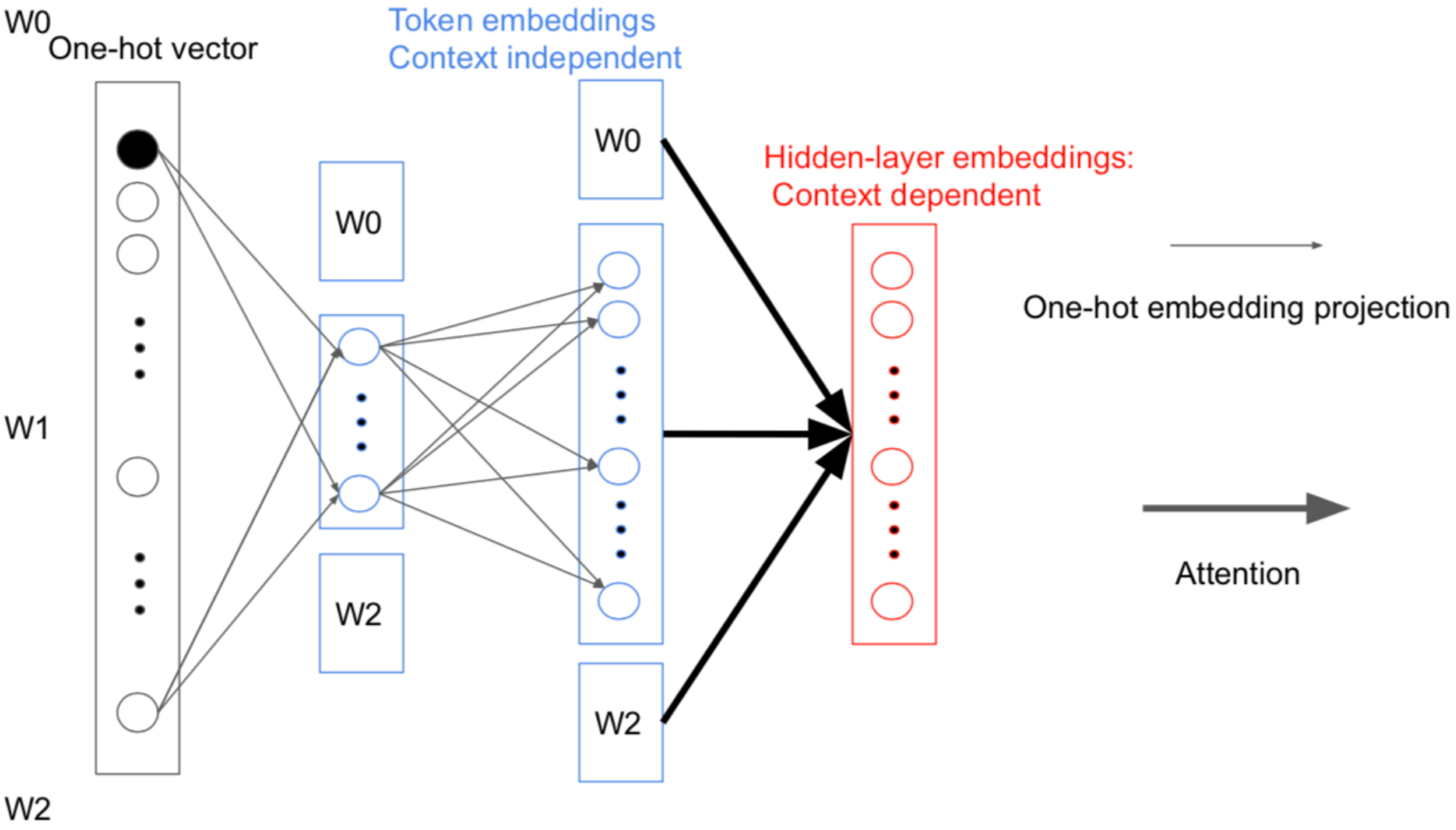
对embedding因式分解

- Token embeddings are context independent while hidden layer embeddings are context dependent.
- Token embeddings are sparsely updated.



O (V x H) → O (V x E + E x H) where E << H


W0
One-hot vector
Token embeddings
Context independent
W0
Hidden-layer embeddings:
Context dependent
One-hot embedding projection
W1
Attention
W2
W2


W0
One-hot vector
W1
W2
Token embeddings
Context independent
W0
W2
W0
W2
Hidden-layer embeddings:
Context dependent
One-hot embedding projection
Attention

Method 1: factorized embedding parametrization

H = 768 for all experiments in this table

Model	E	Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base not-shared	64	87M	89.9/82.9	80.1/77.8	82.9	91.5	66.7	81.3
	128	89M	89.9/82.8	80.3/77.3	83.7	91.5	67.9	81.7
	256	93M	90.2/83.2	80.3/77.4	84.1	91.9	67.3	81.8
	768	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3

-19M

-0.6

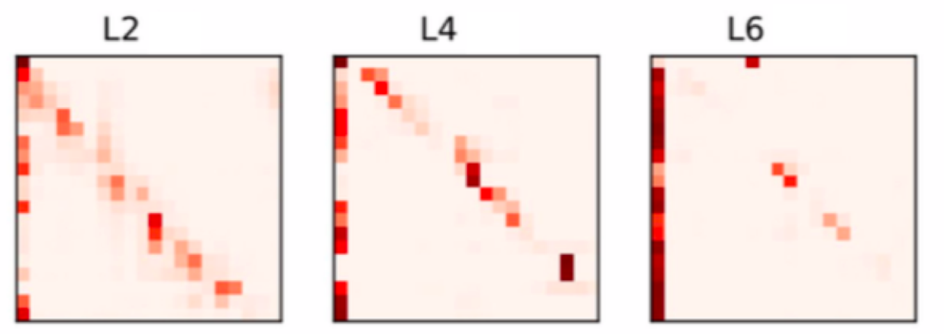
>17% of overall parameter reduction

> 80% of parameter reduction of the projection block

< 1% performance drop

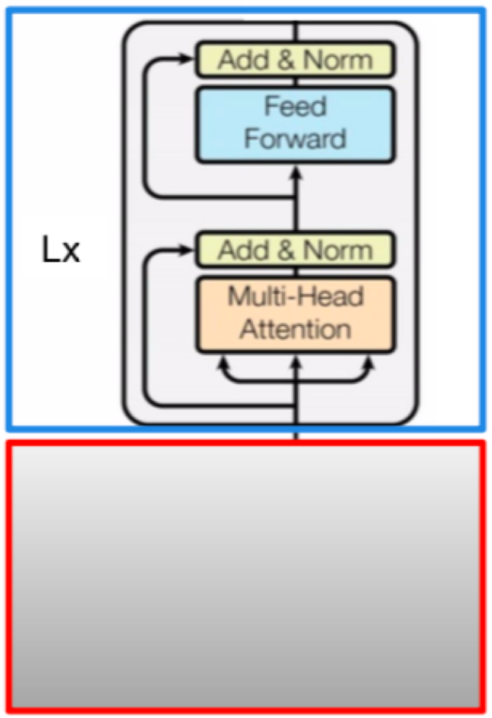
Method 2: Cross-layer parameter sharing

跨层的参数共享



- The attention-feedforward operations are repeated, can we share their parameters?

- Visualization of attention map show similar operation across layers



O (12 x L x H x H) → O (12 x H x H)

Gong, Linyuan, et al. "Efficient Training of BERT by Progressively Stacking." *ICML*. 2019.

Method 2: Cross-layer parameter sharing

Models		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base E=768	all-shared	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8
	shared-attention	83M	89.9/82.7	80.0/77.2	84.0	91.4	67.7	81.6
	shared-FFN	57M	89.2/82.1	78.2/75.4	81.5	90.8	62.6	79.5
	not-share	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base E=128	all-shared	12M	89.3/82.3	80.0/77.1	82.0	90.3	64.0	80.1
	shared-attention	64M	89.9/82.8	80.7/77.9	83.4	91.9	67.6	81.7
	shared-FFN	38M	88.9/81.6	78.6/75.6	82.3	91.7	64.4	80.2
	not-shared	89M	89.9/82.8	80.3/77.3	83.2	91.5	67.9	81.6

-77M (E=768: not-share → all-shared)
-77M (E=128: not-shared → all-shared)
-2.5 (E=768 Avg: not-share → all-shared)
-1.5 (E=128 Avg: not-shared → all-shared)

>70% of overall parameter reduction
> 90% of parameter reduction for the attention-feedforward block

<= 3% performance drop

Method 2: Cross-layer parameter sharing

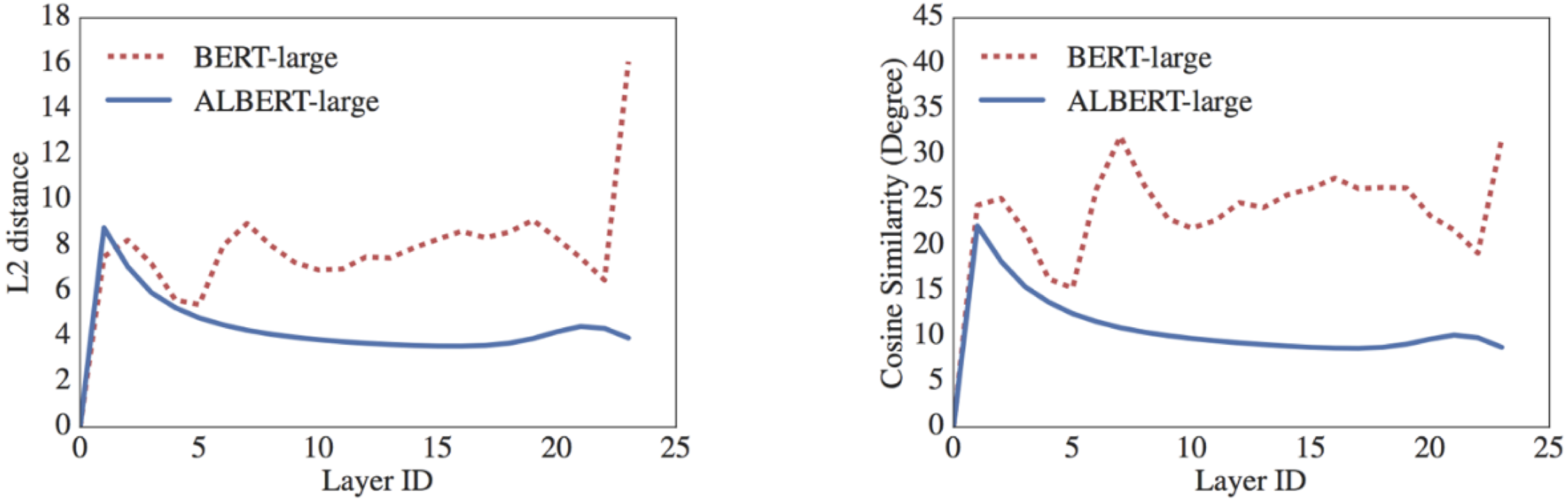
Most of the performance drops are from sharing the parameters of the feed-forward layers

	Models	Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base E=768	all-shared	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8
	shared-attention	83M	89.9/82.7	80.0/77.2	84.0	91.4	67.7	81.6
	shared-FFN	57M	89.2/82.1	78.2/75.4	81.5	90.8	62.6	79.5
	not-share	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base E=128	all-shared	12M	89.3/82.3	80.0/77.1	82.0	90.3	64.0	80.1
	shared-attention	64M	89.9/82.8	80.7/77.9	83.4	91.9	67.6	81.7
	shared-FFN	38M	88.9/81.6	78.6/75.6	82.3	91.7	64.4	80.2
	not-shared	89M	89.9/82.8	80.3/77.3	83.2	91.5	67.9	81.6

0.3

Most of the performance drop come from sharing the parameter of the FFN layer

measurement on the L2 distances and cosine similarity show that our embeddings are oscillating rather than converging



Combining both parameter reduction techniques works best

Method 1: E=768 → E=128
Method 2: not share → all shared

Models		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base E=768	all-shared	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8
	shared-attention	83M	89.9/82.7	80.0/77.2	84.0	91.4	67.7	81.6
	shared-FFN	57M	89.2/82.1	78.2/75.4	81.5	90.8	62.6	79.5
	not-share	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base E=128	all-shared	12M	89.3/82.3	80.0/77.1	82.0	90.3	64.0	80.1
	shared-attention	64M	89.9/82.8	80.7/77.9	83.4	91.9	67.6	81.7
	shared-FFN	38M	88.9/81.6	78.6/75.6	82.3	91.7	64.4	80.2
	not-shared	89M	89.9/82.8	80.3/77.3	83.2	91.5	67.9	81.6

1. > 88% overall parameter reduction
2. < 3.0% absolute average performance drop

Let’s try to scale it up again in width of the network

Model		Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	60M	24	2048	128	True
	xxlarge	235M	12	4096	128	True

An ALBERT-xxlarge model with 4x wider than BERT-large but only has 70% of its parameters

Let's try to scale it up again in width of the network

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3	4.7x
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2	1.0
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1	5.6x
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4	1.7x
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5	0.6x
	xxlarge	235M	**94.1/88.3**	**88.1/85.1**	**88.0**	**95.2**	**82.3**	**88.7**	0.3x

1. 30% parameter reduction
2. 3.5% absolute average performance improvement
3. 3x slower in training

How about increasing depth?

Number of layers	Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
1	18M	31.1/22.9	50.1/50.1	66.4	80.8	40.1	52.9
3	18M	79.8/69.7	64.4/61.7	77.7	86.7	54.0	71.2
6	18M	86.4/78.4	73.8/71.1	81.2	88.9	60.9	77.2
12	18M	89.8/83.3	80.7/77.9	83.3	91.7	66.7	81.5
24	18M	**90.3/83.3**	**81.8/79.0**	83.3	91.5	**68.7**	**82.1**
48	18M	90.0/83.1	**81.8/78.9**	**83.4**	**91.9**	66.9	81.8

A 48-layer ALBERT-large get worse performance than a 24-layer ALBERT-large

How about increasing depth?

Number of layers	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
12	94.0/88.1	88.3/85.3	87.8	95.4	82.5	88.7
24	94.1/88.3	88.1/85.1	88.0	95.2	82.3	88.7

A 12-layer ALBERT-xxlarge get similar performance as a 24-layer ALBERT-xxlarge

NSP：主题预测与关系一致性预测，主题识别影响大

NSP is too easy as a sentence classification task

Sentences are next to each other

1st	google is an american multinational technology company that specializes in internet-related services.
2nd	it is considered one of the big four technology companies, alongside amazon, apple and facebook.

✔

Sentences are from different documents

1st	google is an american multinational technology company that specializes in internet-related services.
2nd	california is a state in the pacific region of the united states.

✘

Sentence order prediction (SOP) as a solution

Sentences are next to each other

1st	google is an american multinational technology company that specializes in internet-related services.
2nd	it is considered one of the big four technology companies, alongside amazon, apple and facebook.

Simply reverse the order

1st	it is considered one of the big four technology companies, alongside amazon, apple and facebook.
2nd	google is an american multinational technology company that specializes in internet-related services.

SOP works better than NSP

SP tasks	Intrinsic Tasks			Downstream Tasks					
	MLM	NSP	SOP	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
None	54.9	52.4	53.3	88.6/81.5	78.1/75.3	81.5	89.9	61.7	79.0
NSP	54.5	90.5	52.0	88.4/81.5	77.2/74.6	81.6	**91.1**	62.3	79.2
SOP	54.0	78.9	86.5	**89.3/82.3**	**80.0/77.1**	**82.0**	90.3	**64.0**	**80.1**

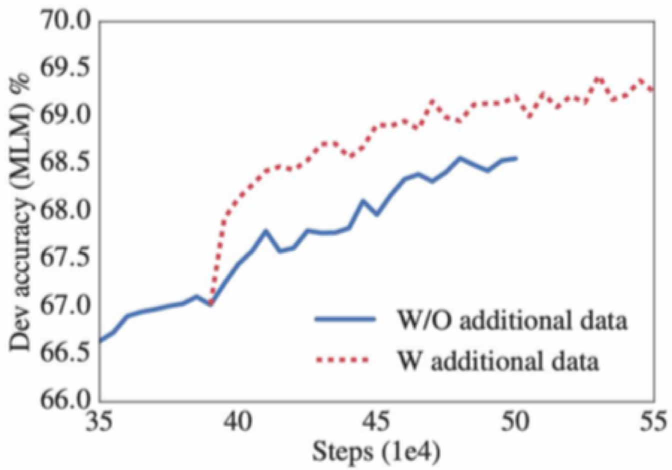
1. NSP model has no ability to predict SOP task
2. SOP model can predict NSP task
3. SOP outperform NSP and no sentence classification task (i.e. None).

Further increase model capacity by removing dropout

	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
With dropout	94.7/89.2	89.6/86.9	90.0	96.3	85.7	90.4
Without dropout	**94.8/89.5**	**89.9/87.2**	**90.4**	**96.5**	**86.1**	**90.7**

Removing dropout consistently improves performance.

Adding 10x more training data slightly increase the performance



	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
No additional data	**89.3/82.3**	**80.0/77.1**	81.6	90.3	64.0	80.1
With additional data	88.8/81.7	79.1/76.3	**82.4**	**92.8**	**66.0**	**80.8**

Increasing training data improves overall performance.

- Improve the parameter efficiency
- Improve self-supervised learning tasks
- Improve model capacity
- Improve training data size

Top on several NLU leaderboard

For about 1 month :)

RACE Leaderboard

Model	Report Time	Institute	RACE	RACE-M	RACE-H
Human Ceiling Performance	Apr. 2017	CMU	*94.5*	*95.4*	*94.2*
Amazon Mechanical Turker	Apr. 2017	CMU	73.3	85.1	69.4
ALBERT (ensemble)	Sept. 26th 2019	Google Research & TTIC	**89.4**	**91.2**	**88.6**
ALBERT	Sept. 26th 2019	Google Research & TTIC	86.5	89.0	85.5
RoBERTa + MMM	Oct. 1st 2019	MIT & Amazon Alexa AI	85.0	89.1	83.3
DCMN+ (ensemble)	Sept. 2019	SJTU & CloudWalk	84.1	88.5	82.3
RoBERTa	July 2019	Facebook AI	83.2	86.5	81.8

GLUE Leaderboard

Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI
1	ALBERT-Team Google Language	ALBERT (Ensemble)		89.4	69.1	97.1	93.4/91.2	92.5/92.0	74.2/90.5	91.3	91.0	99.2	89.2	91.8
2	王玮	ALICE v2 large ensemble (Alibaba DAMO NLP)		89.0	69.2	97.1	93.6/91.5	92.7/92.3	74.4/90.7	90.7	90.2	99.2	87.3	89.7
3	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)		88.8	68.0	96.8	93.1/90.8	92.4/92.2	74.8/90.3	91.1	90.7	98.8	88.7	89.0
4	Facebook AI	RoBERTa		88.5	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	98.9	88.2	89.0
5	XLNet Team	XLNet-Large (ensemble)		88.4	67.8	96.8	93.0/90.7	91.6/91.1	74.2/90.3	90.2	89.8	98.6	86.3	90.4
6	Microsoft D365 AI & MSR AI	MT-DNN-ensemble		87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0
7	GLUE Human Baselines	GLUE Human Baselines		87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.9

SQuAD Leaderboard

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Sep 18, 2019	ALBERT (ensemble model) *Google Research & TTIC* https://arxiv.org/abs/1909.11942	**89.731**	**92.215**
2 Jul 22, 2019	XLNet + DAAF + Verifier (ensemble) *PINGAN Omni-Sinitic*	88.592	90.859
2 Sep 16, 2019	ALBERT (single model) *Google Research & TTIC* https://arxiv.org/abs/1909.11942	88.107	90.902
2 Jul 26, 2019	UPM (ensemble) *Anonymous*	88.231	90.713
3 Aug 04, 2019	XLNet + SG-Net Verifier (ensemble) *Shanghai Jiao Tong University & CloudWalk* https://arxiv.org/abs/1908.05147	88.174	90.702

https://github.com/google-research/ALBERT

THANKS ~